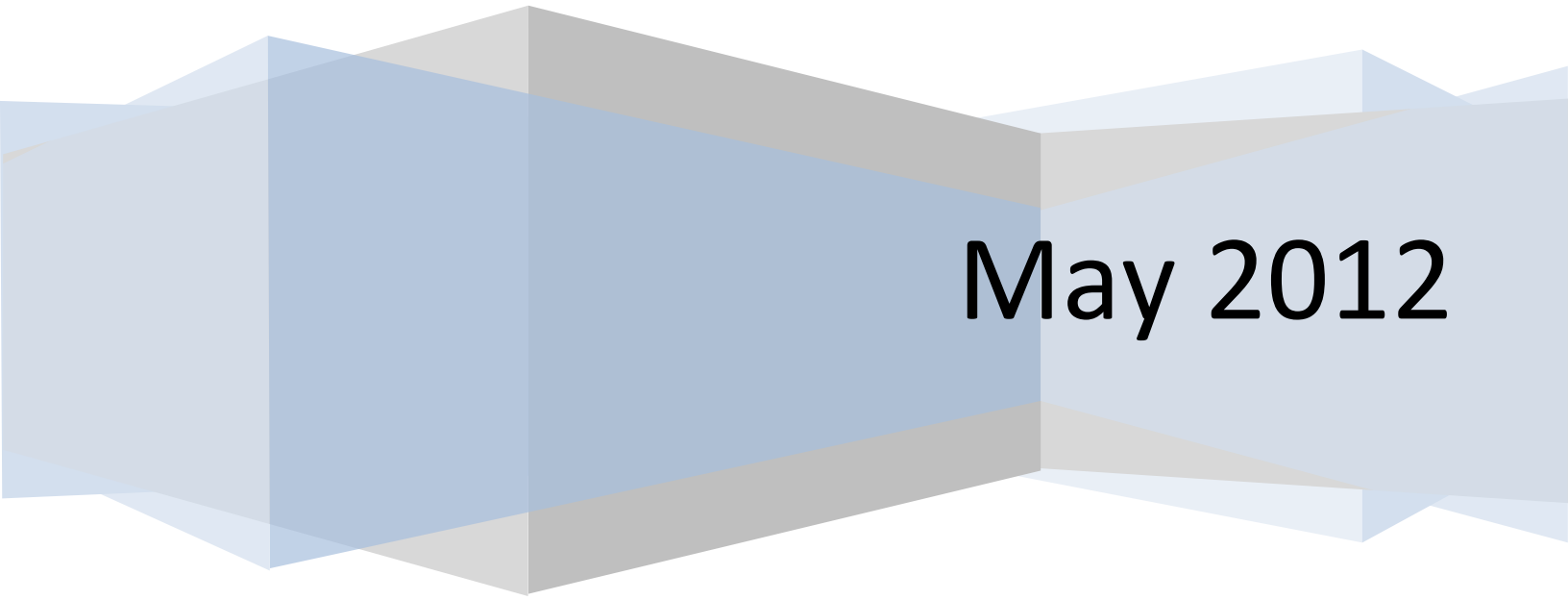
EBSCO Publishing

EBSCO Discovery Service (EDS)
API User Guide



May 2012

EBSCO Discovery Service (EDS) API User Guide

Table of Contents

What is the EDS API? 3
Getting Started 3
Planning Your Access to the EDS API 6
Authentication Method 6
Request Formats 6
Understanding a Search Request 7
Preparing the Search Screen 8
Authenticate, Create Session, and get Info 9
Performing a Search 11
Understanding the Search Results 13
Retrieving a Detailed Record 18
Using the Console to Test Your Queries 21
Using the Console 21

©2012 EBSCO Publishing, all rights reserved.

What is the EDS API?

EBSCO's *Complete Discovery Solution* provides users with access to an institution's entire collection via a single, customizable entry point - creating an experience that is comprehensive, fast, and familiar. With the EDS API, you can integrate EDS directly into your website for a seamless user experience. The EDS API supports REST requests via the HTTP GET and POST methods as well as SOAP requests. Currently there are two available response formats: XML and JSON.

Getting Started

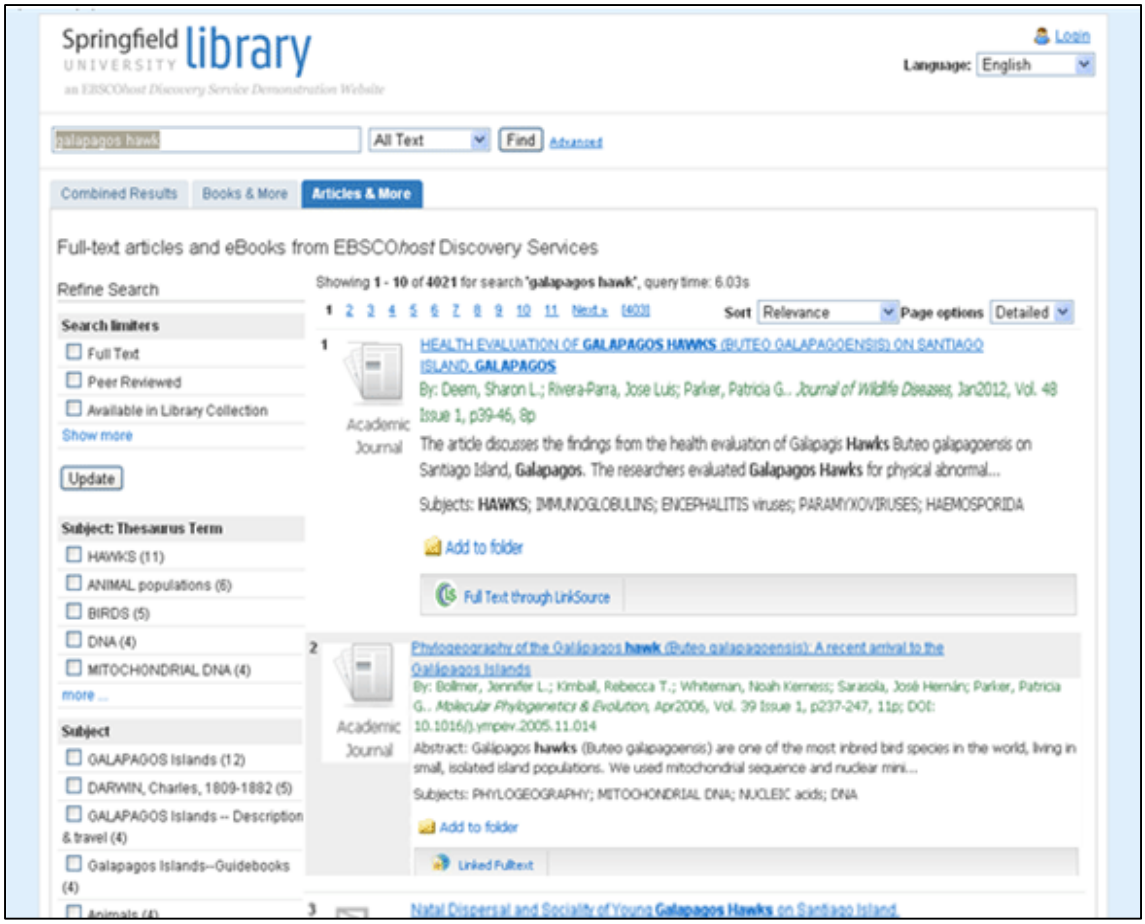
Contact your EBSCO Publishing representative (eds@ebscohost.com) to enable your account for API access. You'll need to know how you authenticate users into your service: by IP address or with your school's UserID and Password.

Think about why you're using the EDS API and how you want to display the results. The API has many options. You will probably use only a few of them. You'll save a lot of time later if you start with a design for your results page.

In the screenshot below, we use an example from our VuFind demonstration of the EDS API. VuFind is a popular open source catalog search for libraries. Our demo has one tab that shows books from the local catalog in the left column and EDS API articles on the right column, plus a separate tab for just catalog items and one for just articles. This screenshot is from the Articles tab, so its results are all articles from EDS. You can also load your catalog directly into EDS and show all results on one screen, with both books and articles mixed together. Choose the display options that fit your library and your user community the best.

This example illustrates some of the key design features that use the EDS API's **Search** method.

©2012 EBSCO Publishing, all rights reserved.



Some of the common UI elements shown in this example are:

- The search box defaults to searching all fields, but can also search just the author, title, subject, or publication, using the **Search** method and the FieldCode parameter.
- Searches can be limited to records in your collection, or only full-text or peer-reviewed articles, or both (plus many more "limiters"), using the Limiters parameter.
- Results are paginated and can be sorted. You can choose the default sort and the number of articles to retrieve per page, using sort and pagesize parameters.
- The title can be underlined to show that users can click on it, taking them to a more detailed page. The full article is returned with the **Retrieve** method.
- "Facets" are returned with the results, showing the most common attributes of the search results. Facets are typically made "clickable", so the user can reduce the search results to only a particular subject, source, type of content, etc. Facets are different for every search, because they're based on the results of that search. We have two different parameters that let developers request a search modified by parameter.

©2012 EBSCO Publishing, all rights reserved.

- Authors and publication source are shown in a different color. This is entirely a decision of the developer or designer. We provide the display value; but we also provide a "searchlink", so you can make these fields clickable and search all your databases for that author or publication.
- Only a few lines of the abstract are shown, and the search term is highlighted in the abstract. You can decide how much of the abstract to show or if you want to skip the abstract. The highlight parameter tells the EDS API whether to highlight the search term in the results.
- Links to full text articles are shown, as well as what type of article it is.

In the rest of this document, we'll cover how to use the EDS API, using this example, and go into more detail on the specific methods used by a screen like this. We'll use the most common approach, the REST API with the HTTP GET protocol. These sections are:

- **Planning Your Access – what formats and protocols will you use**
- **Understanding a Search Request – a brief overview of the methods you'll use**
- **Preparing the Search Screen –actions that happen before the first search**
- **Performing a Search – creating a search**
- **Understanding the Search results – using the detailed XML results**
- **Retrieving a Detailed Record – once the user chooses a record, get its full-text and the rest of its data**

©2012 EBSCO Publishing, all rights reserved.

Planning Your Access to the EDS API

The EDS API supports several access methods. You'll want to decide the following:

- How to authenticate (identify your EBSCO Publishing customer account)
- The search protocol for calling the EDS API (REST or SOAP)
- The format for results (XML or JSON)

Authentication Method

This is how you establish your school's access to EBSCO content. The two most common methods are:

- IP address, if you plan to access the API from server(s) with a fixed IP address.
- UserID, if you might connect to the API from machines outside your network, or from many different addresses.

Request Formats

You can use the EDS API with either the REST or SOAP protocol. REST is a lighter protocol and it is what we'll use in the examples that follow. You can use any EDS API feature through REST or SOAP.

When using REST, you submit requests using either GET or POST. Typically, GET is used to request data, and POST is used to submit data. We'll use GET for these examples, but again, you can choose either one, or a combination (submitting some requests through GET and some through POST). If you're using GET, the search parameters go on the URL sent to the EDS API.

Data is returned from the EDS API in either XML or JSON format. We'll use XML for these examples.

The instructions below apply to both REST and SOAP, but the syntax is different for SOAP. Refer to the EDS API Reference Guide for the SOAP syntax.

©2012 EBSCO Publishing, all rights reserved.

Understanding a Search Request

At its most basic level, a search request requires a query term on the GET request you submit to the EDS API using the **Search** method. Many other options can be added to the **Search** method URL as search parameters. These parameters tell the EDS API what fields you want to search, the desired sort order, how to limit the results, how many results belong on a page, how many fields to include for each search result. Most of the important options have defaults.

The EDS API then returns a block of XML data with your results.

In addition to Search, there are other methods:

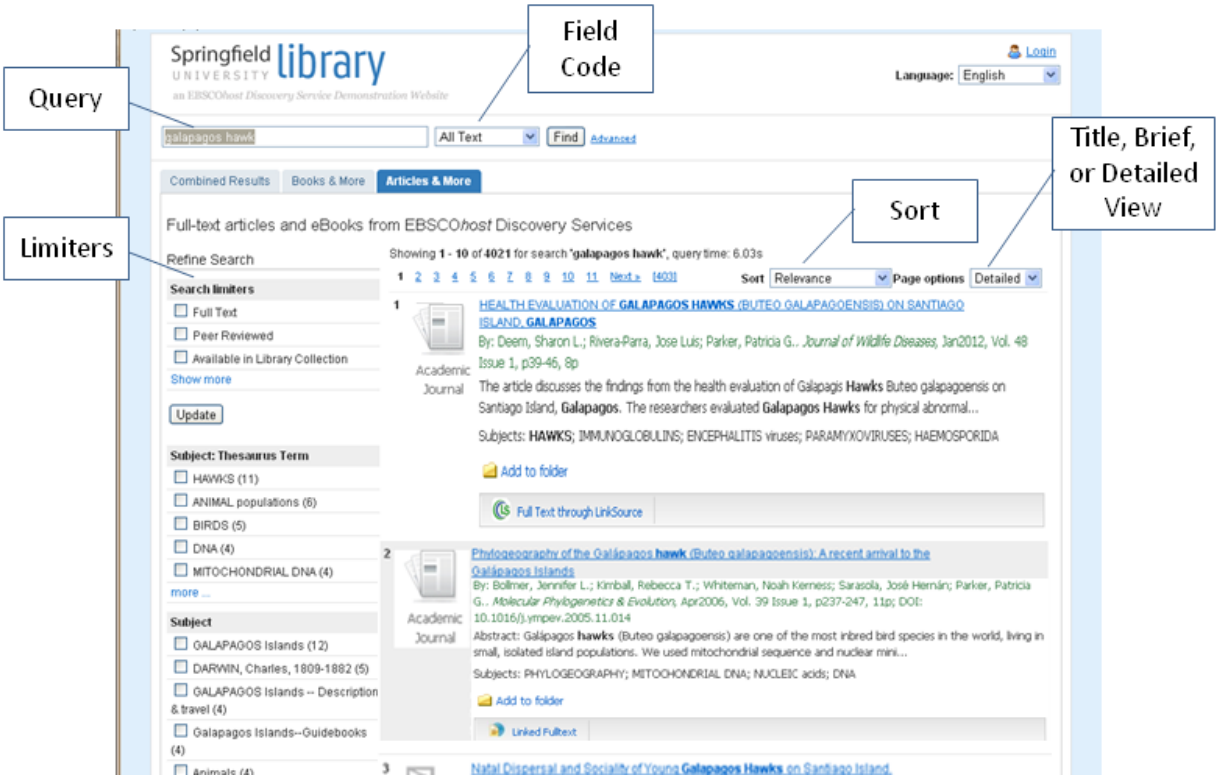
- All requests must include an Authentication Token and a Session ID, so you'll need the **Authenticate** method, the **CreateSession** method, and the **EndSession** method. Authenticate provides a token that must be added to the http header of all subsequent requests. CreateSession provides a SessionToken that must be added to the header of all subsequent requests.
- Some options are defined by your EBSCO customer account in EBSCO*admin*. Use the **Info** method to find out what limiters you have defined for your EBSCO account and which sort options and search fields you can use with your profile. You can use the data returned from the info method to construct a search interface.
- The **Search** method does not return all the fields for each record, or the link to the full-text article. It brings back metadata for each record, so you can display a list of results. When the user selects one particular record, you use the **Retrieve** method to obtain the rest of the data (including Full Text if available) for that article. Often you will want to modify an existing search, such as using facets to reduce the results to a particular author, subject, or source. Rather than simply start over with a modified search, you can use **Actions** to add additional parameters to the existing search, such as facets to reduce the search or pagenumber to go to a specific page. The EDS API will recognize that it's a modification of the same search, and this will save time, and often make your coding easier.

©2012 EBSCO Publishing, all rights reserved.

Preparing the Search Screen

Some search options should be determined before you execute the search, and some are applied after a search, to the results of the search. Let's look at the decisions you make before your first search:

1. What options do you want to provide for search fields, and which one is displayed as default? We call these FieldCodes. They can include author, subject, title, and many others. The **Info** method identifies the FieldCodes available on your account. The default search is a keyword search which searches all these fields, along with the full text if the "Also search within the full text of the articles" expander is selected.
2. What limiters do you want to offer? These can include full-text, peer-reviewed, in your local collection, or dozens of others. Typically, the most commonly-used limiter is full-text, so that the results of a search are all articles to which you have full access in your EBSCO account.
3. What sort options do you want to show, and which one is displayed as the default? Since EDS calculates relevance based on many factors, including date, it is the best default sort.
4. Finally, you can decide if you want to return only the Title, or a Brief view (subjects, authors, source), or a more Detailed view (including the Abstract).



Some schools may "hardwire" some of these options, like Brief View or Sort, and not present them to the users. The options are then added to the search by your code, rather than by any choices made by users. If you use **Info** for limiters and sort options, you ensure that any changes made to your organization's profile in EBSCO*admin* will be reflected in the choices offered to users.

©2012 EBSCO Publishing, all rights reserved.

Before calling **Info**, you need to Authenticate and get a Session ID, so we know which profile you're using and that your server is authorized to access it.

Authenticate, Create Session, and get Info

The endpoint for the Authenticate methods are:

https://eds-api.ebscohost.com/authservice/rest/ipauth
https://eds-api.ebscohost.com/authservice/rest/uidauth

Note: The Authenticate request must be a POST.

The required parameters are:

- UserId
- Password

These are your account credentials from EBSCO*admin*. **Authenticate** will return:

```
<AuthResponseMessage xmlns="http://www.ebscohost.com/services/public/AuthSe
rvice/Response/2012/06/01" xmlns:i="http://www.w3.org/2001/XMLSchema-
instance">
  <AuthToken>5f100347-794c-4d9c-af8a-40943fc4012b</AuthToken>
  <AuthTimeout>1800</AuthTimeout>
</AuthResponseMessage>
```

Save the AuthToken; you will need it for everything else.

The endpoint to create a session is **http://eds-api.ebscohost.com/edsapi/rest/createsession**. You must first set an http header field called x-authenticationToken with the AuthToken you saved from Authenticate.

CreateSession will return:

```
<CreateSessionResponseMessage xmlns="http://epnet.com/webservices/EbscoApi/
Contracts" xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
  <SessionToken>
    8036473a-91cf-435a-92d7-022df3de5313.MoGMwBvTmUwNIHNuQsnyeQ==
  </SessionToken>
</CreateSessionResponseMessage>
```

Save the SessionToken you get back. You'll need it for Info, Search, Retrieve, and EndSession.

The endpoint for **Info** is:

http://eds-api.ebscohost.com/edsapi/rest/info

©2012 EBSCO Publishing, all rights reserved.

Note: The AuthToken must be in the http header x-authenticationToken and the SessionToken must be in a header field x-sessionToken for Info to work.

This is probably a good time to end the session. You won't need to ask for Info again, because it applies to all users, so you can think of this setup session as being one time that serves all users. Each user needs a unique session for searching, because the session takes up a license for those databases with limited licenses and keeps information about the current user session to simplify future searches. Since the Info method doesn't apply to just one user, it is reasonable to create a session for Info, then end that session. You don't need to end the session now, but you need to end it before the user leaves.

The syntax for **EndSession** is:

http://eds-api.ebscohost.com/edsapi/rest/endsession

You'll need the x-sessionToken.

©2012 EBSCO Publishing, all rights reserved.

Performing a Search

Once you've decided which pre-search options to present on the screen, you can ask the user for the search query. A simple search for *Galapagos hawk* can look like this, because the other important parameters are set by default:

http://eds-api.ebscohost.com/edsapi/rest/Search?query-1=AND,galapagos+hawk

It can also look like this:

http://eds-api.ebscohost.com/edsapi/rest/Search?query-1=AND,galapagos+hawk&limiter=FT:y&expander=fulltext&sort=relevance&includefacets=y&searchmode=all&view=detailed&resultsperpage=20&pagenumber=1&highlight=y

These two searches do the exact same thing as there are default values set for all parameters except the query.

Remember, you must include the x-authenticationToken and the x-sessionToken in the http headers. Because the authentication token could easily have expired, you must trap for this response and generate a new token if necessary.

You don't have to get a new sessionToken when you authenticate, because your existing sessionToken is still valid. SessionTokens stay valid as long as the user is active. Remember to end the session when the user leaves, because it is using one of your licenses.

Note: The authentication token is a bit tricky, so it's worth a special note here. The normal process is:

- Execute a search, with the authToken and sessionToken included in the headers.
- If you get error 104, Auth Token Invalid, get a new token using Authenticate, then repeat the search.
- If you get error 108, Session Token missing, get a new SessionToken using CreateSession, then repeat the search.

Some developers will prefer to watch for the expiration time of the token, rather than perform a search with an expired one. You do this by keeping track of when the token should expire, and if it is about to expire, create a new one. You still have to trap for errors, though, including an expired token, just in case.

A list of other error codes is in the EDS API Reference Guide.

©2012 EBSCO Publishing, all rights reserved.

The full list of parameters and their values can be found in the EDS API Reference Guide. The most common ones are listed here:

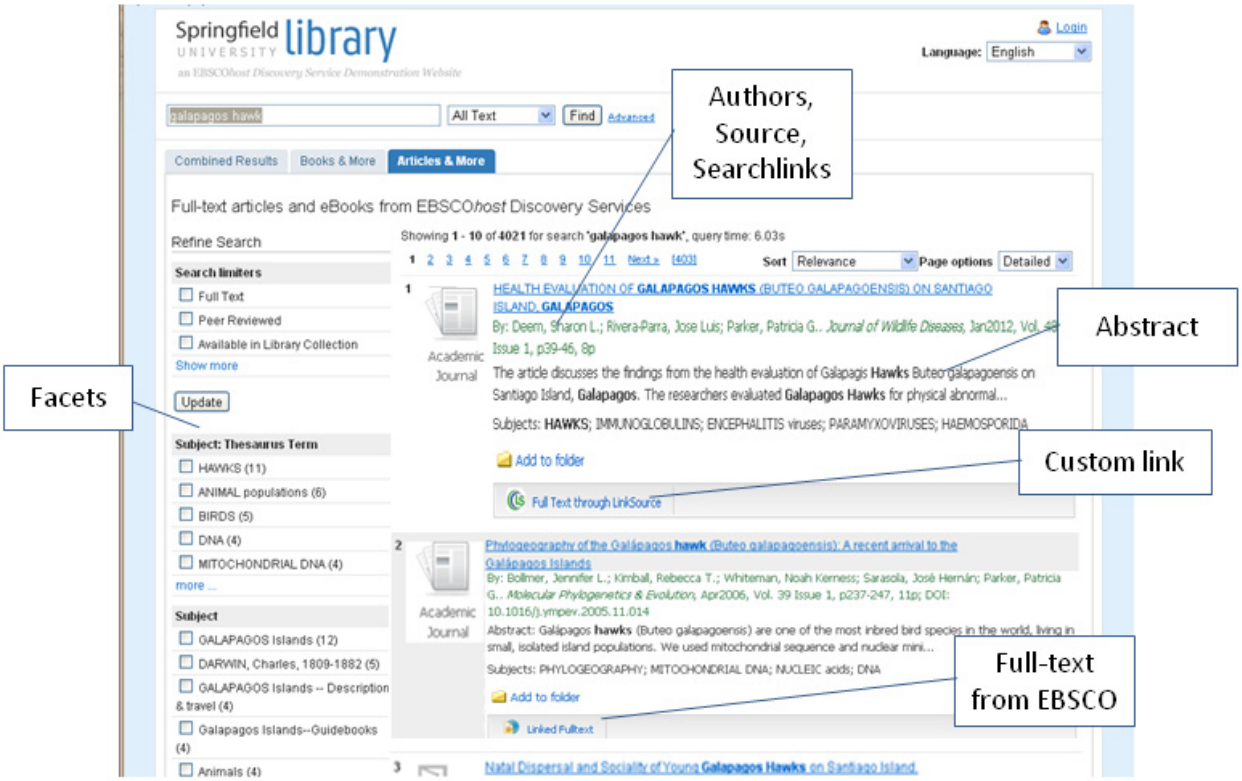
Includefacets	provide available facets in the results	y or n
facetfilter	Apply a facet to refine the search	
Sort	Sort order	Default is relevance; Info determines which sorts are available for your account
Query	Search term, including multiple words and which fields to search	Galapagos hawk, or AU:Elizabeth Moon, or SU:mountaineering; you can have multiple terms and use AND or OR to determine if you want any of those queries or all of them
Limiter	Limiters to apply to the entire search; available limiters are in the results of Info	FT:y, for example, to search only for records with full text available
Searchmode	Search mode (EDS usually uses *all* – find all the words in my query)	Bool, any, all, smart
View	how much detail to show for each result	Title, Brief, or Detailed
Resultsperpage	Page size	Defaults to 20
Pagenumber	Which page to show	Starts with 1, and pagination is done by asking for page 2, 3, etc.
Highlight	Highlight the search term wherever it shows up in the results	Y
Action	Modify the search by adding an action to it, such as to add a facetvalue to the search, resort, or go to a different page	

©2012 EBSCO Publishing, all rights reserved.

Understanding the Search Results

Unless you get an error back, your search results in either XML or JSON will continue all the information you need to identify each record, the pieces of data in each record, and links you can add to your display if you want users to be able to further refine the search. The first section of XML data is information about the query and statistics about the search; the next section contains metadata for every record in the result set, with "items" for each type of metadata for that record; and the last section of XML is the facets and available values for each facet.

Here's an example result list made up of the elements from the response:



The following XML is a small excerpt from a normal results list. There will usually be many examples of these different elements. Some key elements are highlighted below.

```
<SearchResponseMessageGet xmlns="http://epnet.com/webservices/EbscoApi/Contra
cts" xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
<SearchRequestGet>
   <QueryString>
      Query-1=AND,galapagos+hawk
      &sort=relevance&includefacets=n&searchmode=all&view=brief&resultsperpage=
      20&pagenumber=1&highlight=n
   </QueryString>
   <Query>
```


©2012 EBSCO Publishing, all rights reserved.

```
      <BooleanOperator>AND</BooleanOperator>
      <FieldCode/>
      <Term>galapagos hawk</Term>
   </Query>
</SearchRequestGet>
<SearchResult>
   <Statistics>
   <TotalHits>93</TotalHits>
   <TotalSearchTime>1249</TotalSearchTime>
      <Databases>
         <Database>
            <Id>a9h</Id>
            <Label>Academic Search Complete</Lab
            <Status>0</Status>
            <Hits>19</Hits>
         </Database>
         <Database>
            <Id>bth</Id>
            <Label>Business Source Complete</Label>
            <Status>0</Status>
            <Hits>0</Hits>
         </Database>
         … (more databases)
   </Statistics>
   <Data>
      <RecordFormat>EP Display</RecordF
      <Records>
         <Record>
            <ResultId>1</ResultId>
            <Header>
               <DbId>eih</DbId>
               <DbLabel>Environment Complete</DbLabel>
               <An>70483891</An>
               <RelevancyScore>2060</Relevan
            </Header>
            <PLink>
http://search.ebscohost.com/login.aspx?direct=true&db=eih&tg=AN&an=70483891
            </PLink>
            <CustomLinks>
               <CustomLink>

                  <Url>

                  http://atoz.ebsco.com/directLink.6348/OpenURL?sid=EBSCO:eih&gen
                  re=article&issn=00903558&volume=48&issue=1&date=20120101&spage=
                  39&pages=39-
                  46&title=Journal+of+Wildlife+Diseases&atitle=HEALTH+EVALUATION+
                  OF+GALAPAGOS+HAWKS+(BUTEO+GALAPAGOENSIS)+ON+SANTIAGO+ISLAND%2c+
                  GALAPAGOS.&aulast=Deem%2c+Sharon+L.

                  </Url>

                  <Name>A-to-Z as OpenURL (all content)</Name>
```

Number of records found, and total time in milliseconds

Number of results from each of your databases along with a status code (0=success). See the EDS API Reference Guide for status codes.

Beginning of the first result, identified as Record. ResultId is the number of this result in the list of results.

Database and article number for this article. You'll want to display the DbLabel, not the short DbId; if the user wants to see this article in full, or go to full-text, the Retrieve method needs the DbId and the An.

Permanent link to this article. This is often displayed so the user can save or bookmark it, or link into EDS to continue their research.

©2012 EBSCO Publishing, all rights reserved.

```
      <Category>other</Category>

      <Text>findIt@my_Ato-Z</Text>

      <Icon/>

      <MouseOverText>Find full Text through my A-to-Z
      list</MouseOverText>

   </CustomLink>

 </CustomLinks>
<FullText>
   <Links>
      <Link>
         <Type>pdflink</Type>
         <Url/>
         </Link>
   </Links>
   <Text>
      <Availability>1</Availabilit
      <Value/>
   </Text>
<CustomLinks i:nil="true"/>
</FullText>
<Items>
   <Item>
      <Name>Title</Name>
      <Label>Title</Label>
      <Group>Ti</Group>
      <Data>
         HEALTH EVALUATION OF GALAPAGOS HAWKS (BUTEO GALAPAGOENSIS) ON
         SANTIAGO ISLAND, GALAPAGOS.
      </Data>
   </Item>

   <Item>
   <Name>Author</Name>
   <Label>Authors</Label>
   <Group>Au</Group>
   <Data>
        <searchLink fieldCode="AR"
      term="%22Deem%2C+Sharon+L%2E%22">Deem, Sharon L.</searchLink>
      <relatesTo>1</relatesTo> <i> deem@stlzoo.org</i> <br />
      <searchLink fieldCode="AR" term="%22Rivera-
      Parra%2C+Jose+Luis%22">Rivera-Parra, Jose Luis</searchLink>
      <relatesTo>2</relatesTo> <br /> <searchLink fieldCode="AR"
      term="%22Parker%2C+Patricia+G%2E%22">Parker, Patricia
      G.</searchLink> <relatesTo>1,2</relatesTo>
      </Data>
   </Item>
   <Item>
      <Name>TitleSource</Name>
      <Label>Source</Label>
      <Group>Src</Group>
```

The links in the full text section are indicators of the type of full text (most commonly PDF). If PDF is present in the full record, it will be indicated here so you can choose to display the PDF link on your results list. When you use the Retrieve command to get the record the actual link will be returned

Because HTML full text is delivered in the body of the record, its availability indicator is not in the <Links> element but rather in this <Availability> element. A value of 1 means that HTML will be in the full record returned by the Retrieve method

©2012 EBSCO Publishing, all rights reserved.

```
            <Data>
              <searchLink fieldCode="JN"
              term="%22Journal+of+Wildlife+Diseases%22">
              Journal of Wildlife Diseases</searchLink>; Jan2012, Vol. 48
              Issue 1, p39-46, 8p
            </Data>
          </Item>
          <Item>
            <Name>Subject</Name>
            <Label>Subject Terms</Label>
            <Group>Su</Group>
            <Data>
              <searchLink fieldCode="DE"
              term="%22HAWKS%22">HAWKS</searchLink> <br />
              <searchLink fieldCode="DE"
              term="%22IMMUNOGLOBULINS%22">IMMUNOGLOBULINS</searchLink>
              <br /> <searchLink fieldCode="DE"
              term="%22ENCEPHALITIS+viruses%22">ENCEPHALITIS
              viruses</searchLink> <br /> <searchLink fieldCode="DE"
              term="%22PARAMYXOVIRUSES%22">PARAMYXOVIRUSES</searchLink>
              <br /> <searchLink fieldCode="DE"
              term="%22HAEMOSPORIDA%22">HAEMOSPORIDA</searchLink>
            </Data>
          </Item>
          <Item>
            <Name>Abstract</Name>
            <Label>Abstract</Label>
            <Group>Ab</Group>
            <Data>
              The article discusses the findings from the health evaluation
              of Galapagis Hawks B
              Galapagos. The resea
              physical abnormaliti
              transminase values.
              avian encephalitis v
              to haemosporidian an
              Chlamydophila psittaci antigen in a number of Galapagos
              Hawks.
            </Data>
          </Item>
        </Items>
      </Record>
      … (more records on this page)
    </Records>
  </Data>
  <AvailableFacets>
    <AvailableFacet>
      <Id>Journal</Id>
      <Label>Publication</Label>
      <AvailableFacetValues>
        <AvailableFacetValue>
          <Value>condor</Value>
          <Count>8</Count>
          <AddAction>
            addfacetfilter(Journal:condor)
```

Some items include searchLinks, so the information in the data can be used to create a new search. It is encoded so any characters are valid for a new URL. Put the fieldCode:term together as the query, and link the new search to the display value at the end of the searchLink

Display every item in the record, or decide to skip or include some items based on the Group. Detailed view includes those items that we think belong in the detailed view, so you can just include every one of them if you want.

Facets are categories by which these specific search results are grouped, along with the most common results for each facet.

©2012 EBSCO Publishing, all rights reserved.

```
            </AddAction>
          </AvailableFacetValue>
          <AvailableFacetValue>
            <Value>molecular ecology resour
            <Count>6</Count>
            <AddAction>
              addfacetfilter(Journal:molecu
            </AddAction>
          </AvailableFacetValue>
          … (more available facet values fo
        </AvailableFacetValues>
      </AvailableFacet>
      <AvailableFacet>
        <Id>SourceType</Id>
        <Label>Source Type</Label>
        <AvailableFacetValue>
          <Value>Academic Journals</Value>
          <Count>38</Count>
          <AddAction>
            addfacetfilter(SourceType:Academic Journals)
          </AddAction>
        </AvailableFacetValue>
        … (more available facet values for this facet)
    </AvailableFacet>
    … (more facets)
  </SearchResult>
  </SearchResponseMessageGet>
```

There will be many AvailableFacetValues; for example, each journal that shows up in the results list. Typically, you'll only want to display the most common ones. Sort them by Count and display those, along with a link that performs the Search Action addfacetfilter. You can append the addfacetfilter action to your search, to restrict the results to articles

©2012 EBSCO Publishing, all rights reserved.

Retrieving a Detailed Record

Search results, even the detailed view, are only a subset of each record. There is no fixed format for an EBSCO article; since we have hundreds of databases, thousands of publishers, and millions of articles, there is a huge variety of possible formats and data returned in any one article. The **Search** method returns a subset of data in a list; the **Retrieve** method gets all the data for one particular record.

To retrieve a record, you'll need the article number (An) and the database identifier (DbId) from the result list.

The **Retrieve** syntax is

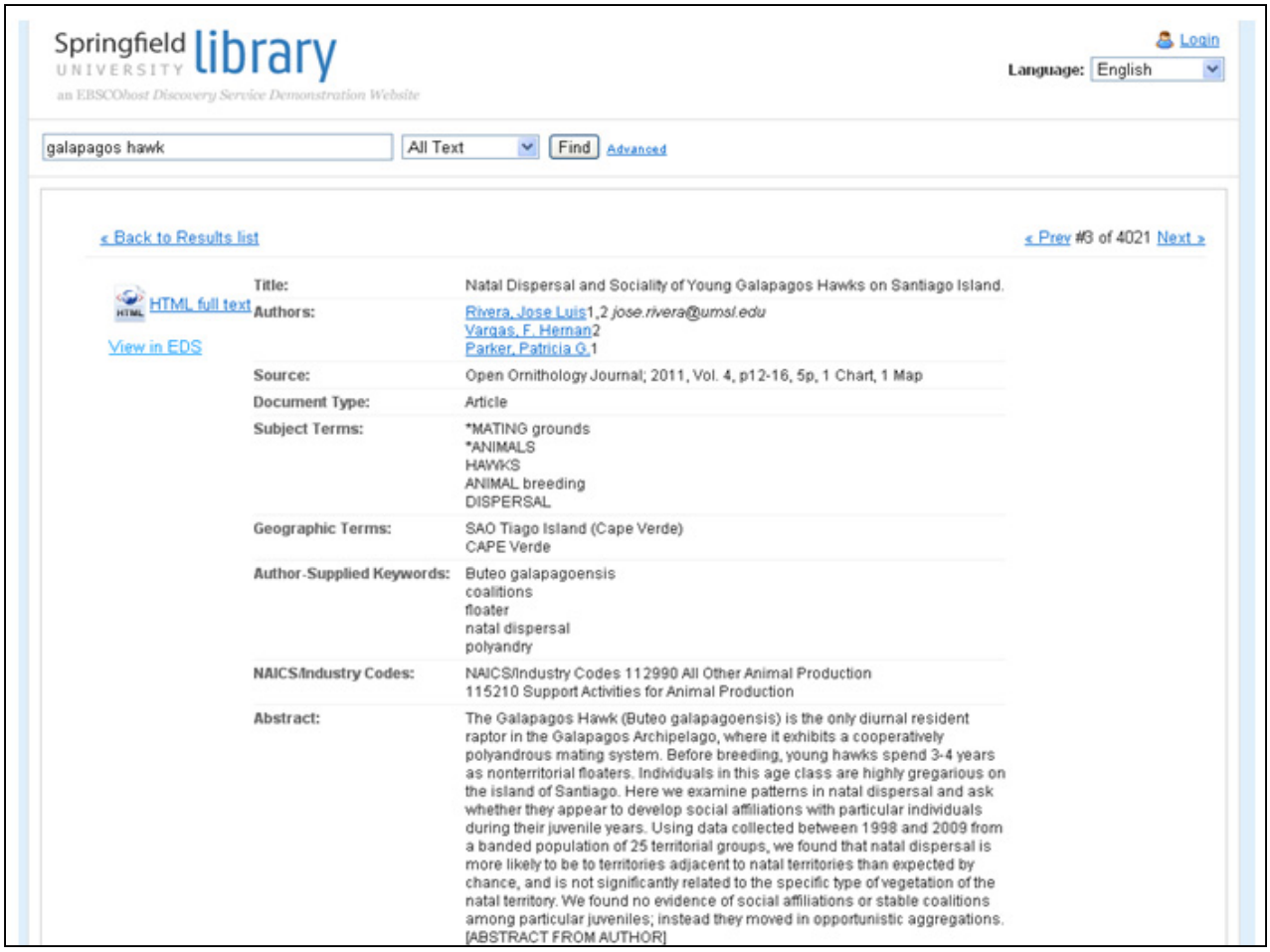
http://eds-api.ebscohost.com/edsapi/rest/retrieve?dbid=a9h&an=36108341

Note: As always, an EDS API request must have the authentication token and session token in the http header. Use the same process as before: do the Retrieve, see if any errors come back, and if the error is an expired token, get a new one before repeating the Retrieve.

The results will be one record with many items in it. Display each item as you get it: first the label, then the value. You don't need to decide which items to include, because all items should be displayed. Typically the order we return items is the order in which you display them, but there might be exceptions: for example, we usually display the database name at the bottom of the list, even though it is returned before any of the items.

Retrieve results are formatted exactly like Search results, except that all metadata fields are returned, the full text element will now have the link to that article, and if there is HTML full text it will be returned for you to display.

©2012 EBSCO Publishing, all rights reserved.


Springfield library
UNIVERSITY
an EBSCOhost Discovery Service Demonstration Website
Login
Language: English
galapagos hawk
All Text
Find
Advanced
« Back to Results list
« Prev #3 of 4021 Next »
HTML full text
View in EDS
Title: Natal Dispersal and Sociality of Young Galapagos Hawks on Santiago Island.
Authors: Rivera, Jose Luis1,2 jose.rivera@umsl.edu
Vargas, F. Hernan2
Parker, Patricia G.1
Source: Open Ornithology Journal; 2011, Vol. 4, p12-16, 5p, 1 Chart, 1 Map
Document Type: Article
Subject Terms: *MATING grounds
*ANIMALS
HAWKS
ANIMAL breeding
DISPERSAL
Geographic Terms: SAO Tiago Island (Cape Verde)
CAPE Verde
Author-Supplied Keywords: Buteo galapagoensis
coalitions
floater
natal dispersal
polyandry
NAICS/Industry Codes: NAICS/Industry Codes 112990 All Other Animal Production
115210 Support Activities for Animal Production
Abstract: The Galapagos Hawk (Buteo galapagoensis) is the only diurnal resident raptor in the Galapagos Archipelago, where it exhibits a cooperatively polyandrous mating system. Before breeding, young hawks spend 3-4 years as nonterritorial floaters. Individuals in this age class are highly gregarious on the island of Santiago. Here we examine patterns in natal dispersal and ask whether they appear to develop social affiliations with particular individuals during their juvenile years. Using data collected between 1998 and 2009 from a banded population of 25 territorial groups, we found that natal dispersal is more likely to be to territories adjacent to natal territories than expected by chance, and is not significantly related to the specific type of vegetation of the natal territory. We found no evidence of social affiliations or stable coalitions among particular juveniles; instead they moved in opportunistic aggregations. [ABSTRACT FROM AUTHOR]

```
<RetrieveResponseMessage xmlns="http://epnet.com/webservices/EbscoApi/Contrac
ts" xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
   <Record>
   <ResultId>1</ResultId>
   <Header>
      <DbId>a9h</DbId>
      <DbLabel>Academic Search Complete</DbLabel>
      <An>36108341</An>
      <RelevancyScore/>
   </Header>
<PLink>
   http://search.ebscohost.com/login.aspx?direct=true&db=a9h&tg=AN&an=36108341
</PLink>
<ImageInfo i:nil="true"/>
<CustomLinks i:nil="true"/>
<FullText>
   <Links>
      <Link>
      <Type>pdflink</Type>
      <Url>
      http://content.ebscohost.com/ContentServer.asp?EbscoContent=dGJyMNLe80Sep
      7U4y9f3OLCmr0qep7JSrq%2B4TbKWxWXS&ContentCustomer=dGJyMODb8nm55eGE4bHjfPH
      X7oEA&T=P&P=AN&S=R&D=a9h&K=36108341
```

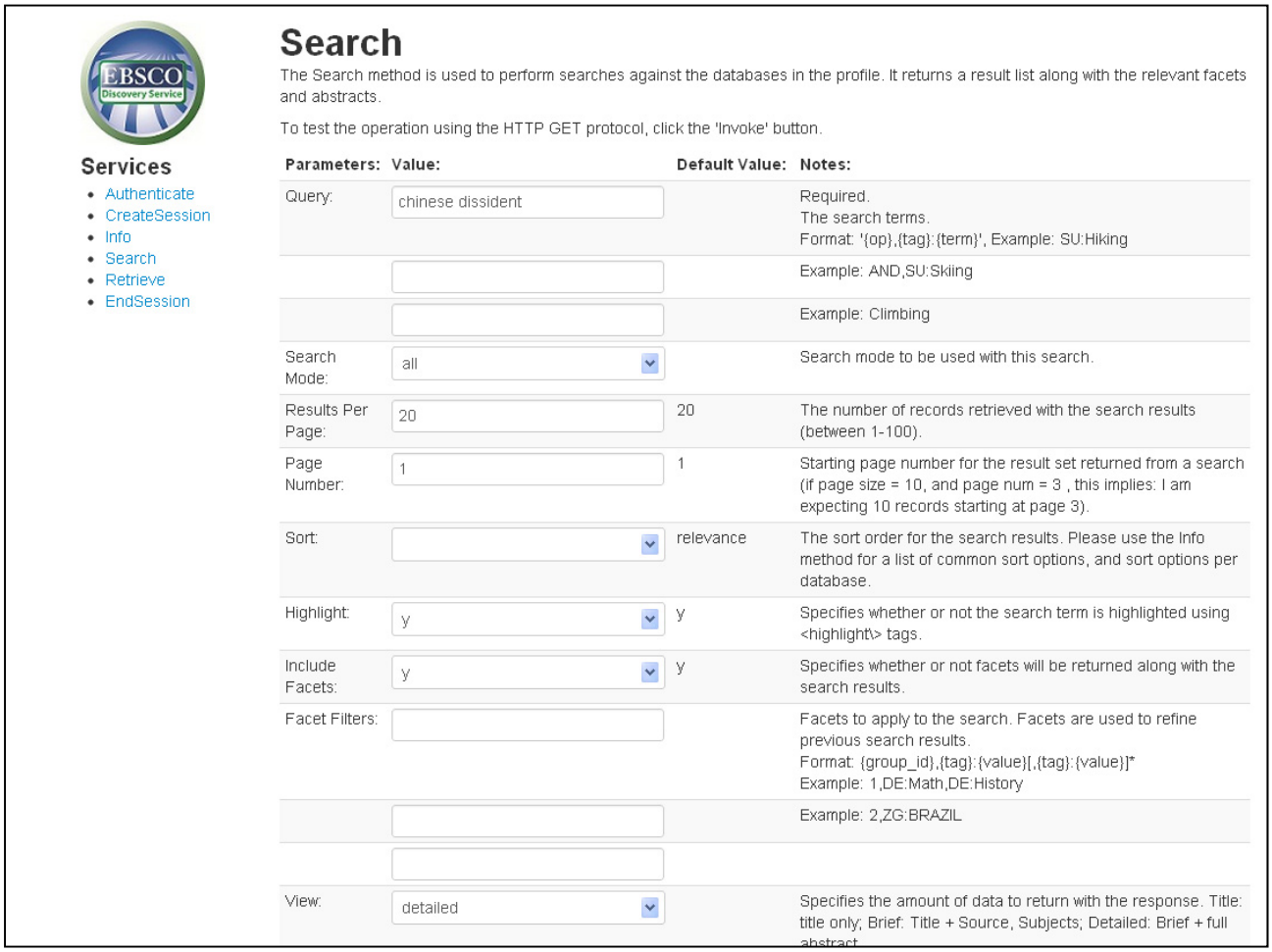

©2012 EBSCO Publishing, all rights reserved.

```
      </Url>
    </Link>
  <Text>
    <Availability>1</Availability>
    <Value>
      <anid>AN0028530099;nhi01feb.08;2008Jan22.08:47;v2.2.460</anid> <jsection
      id="AN0028530099-1">SAMPLINGS</jsection> <title id="AN0028530099-
      2">Emergency Broadcast System </title> <p>Marine iguanas may not be big
      talkers--in fact, they don't vocalize at all--but at least they're good
      listeners. A new study shows that the wary reptiles tune in to the alarm
      calls that GalÃ¡pagos mockingbirds broadcast (for the benefit of their
      own kind) when they spot a common enemy, the GalÃ¡pagos hawk.</p> <p>On
      the island of Santa Fe in the GalÃ¡pagos archipelago, Maren N. Vitousek,
      a graduate student at Princeton University, and three colleagues recorded
      two kinds of mockingbird vocalizations: an alarm call and a territorial
      song. The team then played back both sounds to groups of iguanas. Nearly
      half the iguanas raised their heads or ran away when they heard the alarm
      call. In contrast, fewer than a third reacted to the song.</p> <p>Various
      birds and mammals are likewise known to heed the warnings of other
      species. Those eavesdroppers all make alarm calls of their own, prompting
      biologists to suggest that alarm calls may share acoustical features that
      provoke reactions across species. But the responsiveness of the mute
      iguanas requires other explanations: either the iguanas instinctively
      know what the bird calls mean, or they can learn over time, in Pavlovian
      fashion, that the calls announce the imminent arrival of a dreaded
      predator. (Biology Letters)</p> <p> <img
      src="http://imagesrvr.epnet.com/img/embimages/rdk/nhi/01feb08/12n2.jpg?ep
      host1=dGJyMNLr40Sepq84y9f3OLCmr0qepq5Srqa4SK6WxWXS" alt="12n2.jpg"
      title="Marine iguanas: expert eavesdroppers" /> </p> <aug> <p>By
      StÃ©phan Reebs</p> </aug>
    </Value>
  </Text>
      <CustomLinks i:nil="true"/>
  </FullText>
  <Items>
    <Item>
      <Name>Title</Name>
      <Label>Title</Label>
      <Group>Ti</Group>
      <Data>Emergency Broadcast System.</Data>
    </Item>
    <Item>
      <Name>Author</Name>
      <Label>Authors</Label>
      <Group>Au</Group>
      <Data>
      <searchLink fieldCode="AR" term="%22Reebs%2C+StÃ©phan%22">Reebs,
      StÃ©phan</searchLink>
      </Data>
    </Item>
      … (more display items for this record)
    </Items>
    </Record>
  </RetrieveResponseMessage>
```


©2012 EBSCO Publishing, all rights reserved.

Using the Console to Test Your Queries

There is a simple console application available to help you get started and test the various parameters at https://eds-api.ebscohost.com/Console


Services
Authenticate
CreateSession
Info
Search
Retrieve
EndSession
Search
The Search method is used to perform searches against the databases in the profile. It returns a result list along with the relevant facets and abstracts.
To test the operation using the HTTP GET protocol, click the 'Invoke' button.
Parameters:
Value:
Default Value:
Notes:
Query:
chinese dissident
Required.
The search terms.
Format: '{op},{tag}:{term}', Example: SU:Hiking
Example: AND,SU:Skiing
Example: Climbing
Search Mode:
all
Search mode to be used with this search.
Results Per Page:
20
20
The number of records retrieved with the search results (between 1-100).
Page Number:
1
1
Starting page number for the result set returned from a search (if page size = 10, and page num = 3 , this implies: I am expecting 10 records starting at page 3).
Sort:
relevance
The sort order for the search results. Please use the Info method for a list of common sort options, and sort options per database.
Highlight:
y
y
Specifies whether or not the search term is highlighted using <highlight\> tags.
Include Facets:
y
y
Specifies whether or not facets will be returned along with the search results.
Facet Filters:
Facets to apply to the search. Facets are used to refine previous search results.
Format: {group_id},{tag}:{value}[,{tag}:{value}]*
Example: 1,DE:Math,DE:History
Example: 2,ZG:BRAZIL
View:
detailed
Specifies the amount of data to return with the response. Title: title only; Brief: Title + Source, Subjects; Detailed: Brief + full

Using the Console

Click on the '**Authenticate**' link and enter values:

User ID:
Password:
Response Format: XML or JSON

This will generate the authentication token and display an XML response with that token. You don't need to do anything with this XML response, because the console automatically adds the authToken to a cookie. Use the back button, to get to the previous screen keeping the token in the header. (Cookies must be enabled for the console. The console uses a cookie instead of the http header.)

©2012 EBSCO Publishing, all rights reserved.

Click on the '**Create Session**' link and enter the following values:

Profile ID:
Org ID: (not required)
Guest Session: N
Response Format: XML or JSON

This will generate the session token and display an XML response. You don't need to do anything with this XML response; again, the console will automatically put the sessionToken in the cookie. Use the back button, to get to the previous screen.
Now that both the authentication token and session token have been generated, you can use the **Info**, **Search**, and **Retrieve** methods to see the results

Note! **If the auth / session tokens expire, you will get an XML error message, and you then need to repeat the above steps to get new tokens.**

Click on the '**Info**' link and enter the following values:

Response Format: XML

This will invoke the **Info** method for the above profile and return the available sort options, search tags, expanders and limiters.

Click on the '**Search**' link and enter the following values:

Query: island
Search Mode: all
Page Size: 10
Page Number: 1
Sort: relevance
Highlight: Y
Facets: Y
Facet Filter:
Amount of Data to Return: detailed
Actions:
Limiters:
Expanders:
Response Format: XML

This will invoke the **Search** method for the above profile, and return:

- The complete query and the search criteria applied

©2012 EBSCO Publishing, all rights reserved.

- Search statistics such as total hit count and search time
- Database metadata such as number of hits per database, database names
- For each record returned, the response will display data such as AN number, ISSN, plinks, title, authors, subjects, abstract etc
- Facets related to that result set
- Sort options for that result set

Click on the '**Retrieve**' link and enter the following values:

AN: 55583895
Database: a9h
Highlight: Y
Response Format: XML

This will invoke the **Retrieve** method for the above profile, and return all the data for the requested record, including:

- AN number
- ISSN
- Plinks
- full text
- title
- authors
- subjects
- abstract
- any other data included with that specific article

Click on the '**EndSession**' link and enter the following values:

Session Id: (the session ID will be prefilled)

This will release the session token, releasing any user licenses. For any subsequent searches, a new session token will need to be generated by using the **CreateSession** method.

©2012 EBSCO Publishing, all rights reserved.